TEC-0090

Site Model Based Image Registration and Change Detection

R. Chellappa P. Burlina
A. Rosenfeld C.L. Lin
X. Zhang L.S. Davis

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

November 1996

Prepared for:
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

Monitored by:
U.S. Army Corps of Engineers
Topographic Engineering Center
7701 Telegraph Road
Alexandria, VA 22315-3864

US Army Corps of Engineers
Topographic Engineering Center

T

E

C

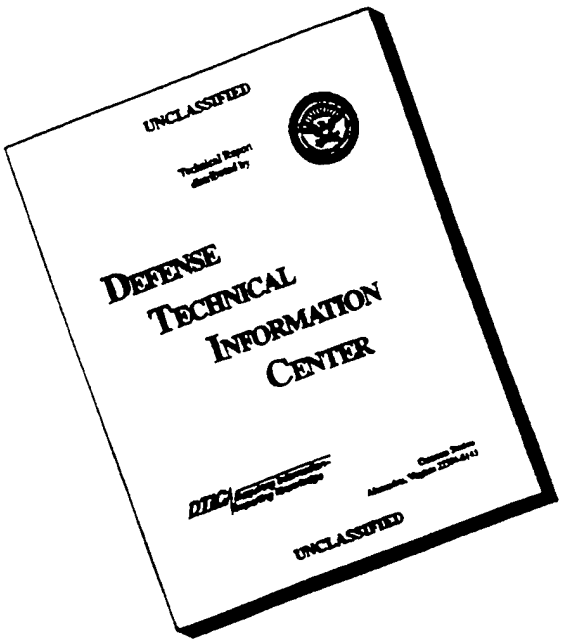
U.S. ARMY TOPOGRAPHIC ENGINEERING CENTER
TEC

Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1996	Technical Sept. 1993 - Sept. 1994

4. TITLE AND SUBTITLE

Site Model Based Image Registration and Change Detection

5. FUNDING NUMBERS

DACA76-92-C-0024

6. AUTHOR(S)

R. Chellappa, P. Burlina, C.L. Lin, X. Zhang, L.S. Davis, A. Rosenfeld

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 N. Fairfax Drive
Arlington, VA 22203-1714

U.S. Army Topographic Engineering Center
7701 Telegraph Road
Alexandria, VA 22315-3864

19. SPONSORING / MONITORING AGENCY REPORT NUMBER

TEC-0090

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The University of Maryland is one of the BAA contractors performing research on aerial image understanding for the RADIUS project. We are contributing model-based and context-based change detection (CD) and monitoring algorithms. Our principal points of focus during the reporting period have been: (a) monitoring movable objects (detection and counting of vehicles), (b) monitoring construction activities for modeled objects, and (c) developing automated image positioning schemes. We exploit geometric and context information derived from site models to accomplish these monitoring tasks. We also have placed special emphasis on the integration of our modules within the RADIUS Common Development Environment (RCDE) platform.

14. SUBJECT TERMS

Change detection, Site Monitoring, Vehicle Counting, Image Positioning

15. NUMBER OF PAGES

47

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

Contents

List of Figures

List of Tables

PREFACE

This report was sponsored by the Defense Advanced Research Projects Agency (DARPA) and monitored by the U.S. Army Topographic Engineering Center (TEC) under contract DACA76-92-C-0024, titled, "Site Model Based Image Registration and Change Detection." The DARPA Program Manager was Dr. Tom Strat, and the TEC Contracting Officer's Representative was Ms. Lauretta Williams.

1 Introduction

The University of Maryland is one of the BAA contractors performing research on aerial image understanding for the Research and Development for Image Understanding Systems (RADIUS) project. We are contributing model-based and context-based change detection (CD) and monitoring algorithms. Change detection (CD) and monitoring, with the goals of locating and identifying significant changes or relevant activities that have occurred between the times of acquisition of the imagery, are core aerial image analysis operations [Strat and Climenson, 1994]. Previous efforts on these applications have relied on general-purpose methods that can be employed to screen a wide variety of imagery, and detect changes without access to any site-specific information. These methods have proven to be unreliable because too many inconsequential changes occur in any natural environment. Therefore, monitoring techniques based on more or less sophisticated differencing of images (possibly after attempted corrections for viewpoint and illumination differences) are extremely sensitive to registration errors and photometric conditions. Even if general-purpose methods could be developed for screening out all changes due to variations in viewpoint and illumination, many differences between the images would still be present, whose significance could only be determined by an image analyst (IA) with comprehensive site knowledge.

The model-based detection schemes presented here incorporate image understanding (IU) techniques whose primitives can be used to direct the system to conduct spatially constrained analyses, whose outcomes may be indicative of occurrences of changes with special intelligence significance. The aerial image exploitation system is site model driven, and is generally based on three classes of primitives: *object primitives*, which correspond to the specific objects that occur in a particular site model, and to the generic object classes supported by the IU system; *spatial primitives*, for the construction of search locales and the specification of constraints on the search for object types within locales; and *temporal primitives*, which can constrain or parameterize the analysis by factors such as time of day, day of week, time of year, etc. This work emphasizes the use of geometric (i.e. object and spatial) primitives. These take the form of site models implemented on the RADIUS RCDE environment [SRI, 1993]. The models encode the spatial relationships between fixed objects of interest in a site,

such as buildings, roads, etc.

IU algorithms designed to extract objects, such as buildings or vehicles in a site for CD applications, cannot be purely bottom-up. The use of site models is instrumental in enabling the incorporation of feedback mechanisms in IU algorithms. For example, in extracting buildings [Venkateswar and Chellappa, 1992], heuristics based on the expected shapes of roofs (site-specific information) can be employed for completing any partial roof hypotheses that result from imperfect bottom-up processing. Likewise, shadow analysis is important for obtaining height information [Huertas and Nevatia, 1988; McKeown, 1990], or allowing the IU system to explain why some building features that are in the field of view cannot be identified in the image. Similarly, site models can provide geometric and photometric constraints that reduce matching ambiguities or search spaces. In sum, the use of site model information increases the reliability and decreases the complexity of IU processes.

In this annual report, various examples of context-based aerial image analysis schemes, illustrating the points made above, are presented—specifically: model-based approaches for (a) automatic and semi-automatic image to site model registration; (b) vehicle detection and counting; and (c) monitoring structured construction activities. In our system, specific approaches have been matched to the tasks at hand: template matching is used for (b) and analytic recognition is used for (c). Site model information is then incorporated in different ways and to various extents according to the selected detection strategy.

The first necessary step of the exploitation cycle within the RCDE involves the registration of an image to the existing site model. Regions of interest (ROIs), depending on the CD task, are subsequently delineated using the site model. Objects, such as buildings and vehicles that are present in the site, are then analyzed for CD, site model refinement, or verification purposes. Careful positioning of newly acquired images is therefore of paramount importance in a model-supported exploitation paradigm. Relevant registration techniques have been described in works such as [Beveridge and Riseman, 1992; Collins et al., 1993; Zheng and Chellappa, 1993]. The emphasis here is on automatic methods. The semi-automatic camera resection algorithm automatically extracts corners corresponding to the intersections of lines. These are chosen as possible image locations of 3D control points. The user can select the correct control point correspondences. Fine location refinement is then

carried out automatically. Camera resection is subsequently accomplished. The fully automatic method uses image-to-image registration and previously resected images to determine the new image positions of control points.

One important facet of our work lies in system integration. One of the principal thrusts behind this research has been the development of IU capabilities focused on the needs of imagery analysts (IAs). To this end, the mechanisms previously described have been developed within or around the RCDE platform, and make substantial use of its integrated functionalities: It provides a common development environment on which RADIUS-related algorithms can be developed and tested. This platform allows for the creation and manipulation of CAD-like objects used to model site objects. It also provides a system from which an operational RADIUS testbed system is derived.

The operational cycle involves the following exploitation scenario: new images are acquired, and they are registered to the existing site model. Images are then prioritized for exploitation. The image analyst launches a set of batch detection algorithms for which a minimal number of parameters need to be specified. All remaining site information is fetched from site objects residing on a database. The successive steps involved in the exploitation cycle—registration, resection, enhancement, detection—are implemented in or around the RCDE.

Section 2 gives a detailed overview of steps preliminary to site-model based CD, as well as the specific type of site model information used by each CD process described subsequently. The semi- and fully-automatic image to site model registration schemes are presented in Sections 3 and 4. Section 5 presents methods for the detection of vehicles in designated areas, such as parking lots and roads. Experimental results are reported in each of these sections. Some specific issues involved in the integration of these algorithms are given greater consideration in Section 7.

2 Site Model Supported Monitoring

A core component of model-based aerial image exploitation is the availability of site models. A site model contains a geometric description of the site under scrutiny, and of the relevant

site features (areas, buildings, roads, etc.). It also includes imaging and photometric parameters associated with available images along with collateral and auxiliary information [ARPA, 1993]. Typical auxiliary information associated with a site model includes: a) an overview of the site, b) a baseline description, and c) comments and analysis tasks. Site model construction requires several overlapping coverage images of the same site to be available, and is carried out under RCDE.

When a newly acquired image becomes available, its registration to the existing site model is a necessary condition for model-based exploitation. As mentioned previously, depending on the particular exploitation task, e.g., if building or vehicle related activity is being monitored, we can use the site model and viewing direction of the new image to identify regions in the image that need further analysis. We can subsequently invoke the necessary IU algorithms related to detection of construction activities, vehicle location and counting (and road extraction, if construction of roads is being monitored).

Registration is accomplished when the exterior orientation parameters aligning the camera frame of reference with the world frame of reference have been carefully determined. The exterior orientation of the camera is specified using the conformal transformation commonly used in photogrammetry. Using the conformal representation, camera-centered coordinates are represented by the position of the camera center and camera orientation in the world coordinates, respectively specified by the coordinates (x_o, y_o, z_o), and rotation angles ω, ϕ, κ. Denoting by (x_c, y_c, z_c) and (x_w, y_w, z_w), the coordinates of a point in the camera centered coordinate system, and in the world coordinate system, the transform from (x_w, y_w, z_w) to (x_c, y_c, z_c) is given by the familiar expression [Moffitt and Mikhail, 1980]

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \mathbf{R}_z(\kappa)\,\mathbf{R}_y(\phi)\,\mathbf{R}_x(\omega) \begin{pmatrix} x_w - x_o \\ y_w - y_o \\ z_w - z_o \end{pmatrix}$$

$$= \mathbf{R} \begin{pmatrix} x_w - x_o \\ y_w - y_o \\ z_w - z_o \end{pmatrix} \tag{1}$$

where

$$\mathbf{R} = \begin{pmatrix} \cos\phi\cos\kappa & \sin\omega\sin\phi\cos\kappa + \cos\omega\sin\kappa & -\cos\omega\sin\phi\cos\kappa + \sin\omega\sin\kappa \\ -\cos\phi\sin\kappa & -\sin\omega\sin\phi\sin\kappa + \cos\omega\cos\kappa & \cos\omega\sin\phi\sin\kappa + \sin\omega\cos\kappa \\ \sin\phi & -\sin\omega\cos\phi & \cos\omega\cos\phi \end{pmatrix}$$

and we also have

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \mathbf{R}^t \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + \begin{pmatrix} x_o \\ y_o \\ z_o \end{pmatrix} \tag{2}$$

We have developed semi- and fully-automated image-to-site-model registration procedures as described in Sections 3 and 4. These procedures enable the determination of image point positions of points whose 3D locations are known (control points). Once conjugate points have been identified, resection can be accomplished to find the above exterior camera orientation parameters.

Perhaps one of the most obvious ways in which site models are used is in the delineation of ROIs. Given an image to be exploited, locating the regions of interest according to the task narrows the search area and reduces computation and false alarms. Delineation not only decreases complexity, but also ensures the success of algorithms which otherwise might sometimes fail. Finally, using context in delineation allows for recognition by function, since location is a determinant factor in the ultimate purpose or function of an object. This is especially critical in applications where target signatures do not allow for easy object discrimination, such as in SAR images. The delineation process is quite straightforward in nature: When a 3D region object is available from the site model, we directly project the region boundaries onto the image to be monitored and label the region(s) in the image domain. The method uses camera model information available from the site model. The delineated region is then further cropped for possible shadows and/or occlusions, again exploiting site model information.

In addition to using site and camera information to infer and delineate ROIs, the bulk of the site model information used by each detection algorithm is geometric in nature. Vehicle dimensions and orientation are used to infer template dimensions and orientation in the local vehicle detection scheme. The selection of templates can be decided from knowledge of the

parking lot occupancy (in a full parking lot, the vehicle sides are likely to be occluded). This could be in the form of priors using context information, or this information could be fed back from the global detector, a possibility currently under study. Information on climatic/weather conditions is not exploited here, but evidently could explain the strengths of brightness gradients; this could be used either for tuning purposes, or to quantify the confidence in the detected changes. The availability of CAD-like models for the visible objects allows for the application of syntactic object recognition techniques. The illumination direction is used along with camera parameters and object description in the construction monitoring module to check for the presence of shadows corroborating the detected vertical lines. Contextual information is also being used in the RCDE system to guide the application of exploitation algorithms: Conditions for which an algorithm's performance is acceptable can be encoded as rules embedded in a production system encoded in Prolog [Strat, 1995]. Context information is then used to trigger the application of the algorithm deemed most appropriate for the task at hand.

3 Semi-Automatic Image to Site Model Registration

Precise image to site model registration is critical to context-based image exploitation. The techniques we are reporting here aim to minimize human intervention as much as possible. The following two sections describe semi- and fully-automatic methods for image-to-site-model registration for images with arbitrary orientation. The semi-automatic camera resection algorithm automatically extracts feature points of structural significance, from which the user can select the correct point correspondences from the projection of the site model. While the user is selecting point correspondences, our system can help decide the precise locations so that the user does not need to carefully pin down positions of points. Camera resection can be accomplished after correspondence relations are established.

3.1 Principle of Operation

While it is often easy for humans to approximately locate control points to within a few pixels, the fine tuning of the locations of the control points takes excessive effort and is

subject to errors. On the other hand, it is relatively easy to devise automated schemes to localize corners with high accuracy, but selecting the ones corresponding to control points is a complex task to automate. The semi-automatic camera resection algorithm presented here automatically extracts the possible locations of control points and lets the user select the control point correspondence. The steps involved in this approach are as follows: (a) Project the existing site model (wire frame and visible control points) into the new image domain using the given approximate camera model. (b) Extract edges and corners (line intersections) from the new image. (c) Manually select the extracted control points and relate them to the corresponding corners; during operation, the corner closest to the mouse position is selected. (d) After at least four pair correspondences are made, camera resection is performed to get the correct camera model for the new image [Methley, 1986].

3.2 Experiments

Figures 1 and 2 illustrate an example of semi-automatic image-to-site-model registration. Figure 1.a shows the initial site-model projection using the approximate camera model, and Figure 1.b shows the corners extracted from the new image. Figure 2.a shows the conjugate points matched with the assistance of the user. Finally, Figure 2.b shows the final registration result. The image registration accuracy can be verified by projection of the site model onto the new image. Table 1 provides results and comparisons between semi-automatic and manual image to site model registration on some RADIUS model-board-2 images. Results show that a slight performance improvement can be expected from the semi-automatic method, as is evident from the lower residual RMS pixel errors.

4 Automatic Multi-Resolution Image-to-Site-Model Registration

4.1 Principle of Operation

We present now an automatic, multiresolution, image-to-site-model registration method. This procedure is mediated by an image-to-image registration scheme.

The image-to-site-model registration method presented is based on an automatic, multi-resolution, image-to-image registration scheme. Image features derived from wavelets are

used in a multi-resolution matching and refinement scheme assuming affine and projective transformations to determine and refine image locations of known 3D control points. Space resection is subsequently carried out to accomplish precise image positioning.

We use a previously registered image, which we denote by O_1^0, where the superscript denotes the resolution, and the subscript, the transformation. The newly acquired image is denoted by N_1^0. A feature detection algorithm based on Gabor wavelets for detecting local curvature discontinuities [Manjunath, 1992] is used to extract features on the old image. The 3D locations of detected feature points can be computed from the known old image camera model. We select features lying on the same plane (here the ground plane) for subsequent analysis. These features are our control points.

Next, we establish a correspondence between these 3D control points and their 2D locations in the new image N_1^0. This consists of the following steps. The exact exterior orientation parameters of the old image, and approximate parameters of the new image, are used to infer an initial projective transformation T_1. Through this transformation, the old image, and the 2D locations of the control points in the old image, are brought into the new image frame. This transformed image is denoted by O_2^0. The resolutions of the original images O_2^0 and N_1^0 are thereupon decreased by a factor of 2^k; the resulting images are denoted by O_2^k and N_1^k. A low resolution area correlation algorithm is first used to establish approximate correspondence of these 2D points between O_2^k and N_1^k. A similarity transformation between O_2^k and N_1^k is computed from initial point matches. The old image is transformed through this similarity, a new match point set is produced, and the feature matches are subjected to a consistency check. Geometrically inconsistent candidate matches are rejected. Using these output feature pairs, a multi-resolution matching refinement is applied using an assumed eight-parameter projective transformation between the two images.

The overall matching procedure is reiterated at various resolution levels so as to arrive at a registration of the two images with an acceptable degree of accuracy. As a result of image-to-image registration, the 2D locations of control points in the new image are available, and resection can be carried out.

Feature Point Detection Image-to-Image registration requires establishing some correspondence between two images. The points used in this correspondence are called feature points. For feature point extraction, a Gabor wavelet decomposition and local scale interaction based algorithm reported in [Manjunath, 1992] is used. The basic wavelet function used in the decomposition is of the form

$$\begin{aligned} \Phi(X,Y,\vartheta) &= e^{-(X'^2+Y'^2)+i\pi X'} \\ X' &= X\cos\vartheta + Y\sin\vartheta \\ Y' &= -X\sin\vartheta + Y\cos\vartheta \end{aligned} \tag{3}$$

where ϑ is the preferred spatial orientation. In our experiments ϑ is quantized into four orientations. The feature points are extracted as the local maxima of the energy measure

$$I(X,Y) = \max_{\vartheta}\{||W_{j_1}(X,Y,\vartheta) - \gamma W_{j_2}(X,Y,\vartheta)||\} \tag{4}$$

with

$$W_j(X,Y,\vartheta) = \mathbf{f} * \Phi(2^{-\frac{j}{2}}X, 2^{-\frac{j}{2}}Y, \vartheta),$$

where j_1 and j_2 are two dilation parameters, and $\gamma = 2^{(j_1-j_2)}$ is a normalizing factor. In implementing the above algorithm, we further require the energy measure for a feature point to be maximum in a neighborhood with radius equal to 10, and also to be above a threshold. Before performing image-to-image registration, the feature point detection algorithm is applied to the old image O_1^0, and the old image along with the detected feature points can be transformed, approximately, into the new image domain.

Old to New Image Transformation Using the exact camera model of the old image, and the approximate camera model of the new image, the old image is transformed, approximately, into the new image domain by an eight-parameter projective transformation [Wolberg, 1988]. Denote this transform by T_1. In the case of remotely-sensed images, the plane transformations between two images are sufficient. Because object heights are very small compared to the camera range (for example, in the RADIUS model-board-2 image set, the camera range is 10,850 feet, and the tallest building is 47.5 feet high, which corresponds to 0.43 percent of the camera range), for practical purposes, most control points will be

considered to be coplanar. Furthermore, for our registration method, this transformation is indeed exact, given that only coplanar control points are chosen. Let (x_i, y_i, z_i) and (X_i, Y_i) respectively, denote a 3D point and its corresponding 2D point in image i domain, and let f_i be the focal length; then, for two cameras characterized by

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \mathbf{R}_i \begin{pmatrix} x_w - x_o^i \\ y_w - y_o^i \\ z_w - z_o^i \end{pmatrix}$$
$$= \begin{pmatrix} r_{11}^i & r_{12}^i & r_{13}^i \\ r_{21}^i & r_{22}^i & r_{23}^i \\ r_{31}^i & r_{32}^i & r_{33}^i \end{pmatrix} \begin{pmatrix} x_w - x_o^i \\ y_w - y_o^i \\ z_w - z_o^i \end{pmatrix}, \quad i = 1, 2$$

and using the coplanarity constraint,

$$Ax_w + By_w + Cz_w = 1 \tag{5}$$

we obtain the familiar projectivity transformation (see for example [Tsai and Huang, 1984; Tsai and Huang, 1981])

$$X_2 = \frac{a_1 X_1 + a_2 Y_1 + a_3}{a_7 X_1 + a_8 Y_1 + 1} \tag{6}$$

$$Y_2 = \frac{a_4 X_1 + a_5 Y_1 + a_6}{a_7 X_1 + a_8 Y_1 + 1} \tag{7}$$

where the projectivity parameters are expressed as functions of the plane parameters along with the interior and exterior orientation parameters characterizing both images, i.e.

$$a_1 = \frac{f_2}{f_1} \frac{r_{11} + \frac{\delta x_o}{D}(Ar_{11}^1 + Br_{12}^1 + Cr_{13}^1)}{r_{33} + \frac{\delta z_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}; \quad a_2 = \frac{f_2}{f_1} \frac{r_{12} + \frac{\delta x_o}{D}(Ar_{21}^1 + Br_{22}^1 + Cr_{23}^1)}{r_{33} + \frac{\delta z_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}$$

$$a_3 = f_2 \frac{r_{13} + \frac{\delta x_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}{r_{33} + \frac{\delta z_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}; \quad a_4 = \frac{f_2}{f_1} \frac{r_{21} + \frac{\delta y_o}{D}(Ar_{11}^1 + Br_{12}^1 + Cr_{13}^1)}{r_{33} + \frac{\delta z_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}$$

$$a_5 = \frac{f_2}{f_1} \frac{r_{22} + \frac{\delta y_o}{D}(Ar_{21}^1 + Br_{22}^1 + Cr_{23}^1)}{r_{33} + \frac{\delta z_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}; \quad a_6 = f_2 \frac{r_{23} + \frac{\delta y_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}{r_{33} + \frac{\delta z_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}$$

$$a_7 = \frac{1}{f_1} \frac{r_{31} + \frac{\delta z_o}{D}(Ar_{11}^1 + Br_{12}^1 + Cr_{13}^1)}{r_{33} + \frac{\delta z_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}; \quad a_8 = \frac{1}{f_1} \frac{r_{32} + \frac{\delta z_o}{D}(Ar_{21}^1 + Br_{22}^1 + Cr_{23}^1)}{r_{33} + \frac{\delta z_o}{D}(Ar_{13}^1 + Br_{23}^1 + Cr_{33}^1)}$$

with

$$D = 1 - (Ax_o^1 + By_o^1 + Cz_o^1)$$

The above results follow from straightforward algebraic derivations (see for example [Tsai and Huang, 1984; Tsai and Huang, 1981]). This result can also be derived using projective geometry; the interested reader is referred to [Faugeras, 1993]. These eight parameters exactly describe the plane-to-plane transformation between the two images. Three rotation parameters and three translation parameters specify the transformation between the coordinate systems of camera-1 and camera-2. f_1 specifies the perspective projection between camera-1 and image-1, and f_2 the perspective projection between camera-2 and image-2. Therefore, eight parameters can describe the plane transformation between image-1 and image-2.

The 3D locations of the feature points detected in the old image are calculated from the known camera model of the old image and the Digital Terrain Model (DTM) in the site model. Only coplanar 3D points are chosen as control points. The coefficients of the plane equation (5) can be calculated from these coplanar control points. The eight parameters (a_1 to a_8) of transformation T_1 between O_1^0 and N_1^0 are then computed and O_1^0 is transformed into the coordinate system of N_1^0 using equation (6) and (7). The resulting image is denoted by O_2^0.

Initial Matching To accomplish initial matching, the resolution of images O_2^0 and N_1^0 is first reduced to the lowest level; these low resolution images being denoted by O_2^k and N_1^k, respectively. If the difference in the orientation between the two images is too large, a low resolution matching using simple transform parameters needs to be applied first. At low resolution, a similarity transformation can adequately explain the difference between two images. Here, the four-parameter transformation, including scale, rotation and translation, is used,

$$\begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} = s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} + \begin{pmatrix} \delta X \\ \delta Y \end{pmatrix} \tag{8}$$

where s is the scaling parameter, θ the rotation angle and $(\delta X, \delta Y)^t$ the translation between the two images.

Initial matching is first determined by the best pairwise fit of the feature points detected on image O_2^k in image N_1^k. These matching pairs are then used to estimate the four transform parameters.

Feature Points Matching For a detected feature point in image-1, its match is sought in the corresponding window area in image-2. Let $(2w_s+1)^2$ denote the search window size, $f_1(m,n)$, a feature point in image-1, and $f_2(u,v)$ a feature point in image-2; their mutual correlation coefficient is given by

$$\psi_{f1f2}(m,n;u,v) = \frac{1}{\sigma_1\sigma_2(2w_m+1)^2}\sum_{i,j=-w_m}^{i,j=w_m}(f_1(m+i,n+j)-\mu_1)(f_2(u+i,v+j)-\mu_2) \quad (9)$$

where

$$\mu_1 = \frac{1}{(2w_m+1)^2}\sum_{i,j=-w_m}^{i,j=w_m} f_1(m+i,n+j)$$

$$\mu_2 = \frac{1}{(2w_m+1)^2}\sum_{i,j=-w_m}^{i,j=w_m} f_1(u+i,v+j)$$

$$\sigma_1 = \left(\frac{1}{(2w_m+1)^2-1}\sum_{i,j=-w_m}^{i,j=w_m} f_1(m+i,n+j)^2-\mu_1^2\right)^{\frac{1}{2}}$$

$$\sigma_2 = \left(\frac{1}{(2w_m+1)^2-1}\sum_{i,j=-w_m}^{i,j=w_m} f_1(u+i,v+j)^2-\mu_2^2\right)^{\frac{1}{2}}$$

This mutual correlation coefficient is employed as matching criterion, and feature matching is accomplished between the images O_2^k and N_1^k using this criterion.

Estimation of the Transform Parameters Since the Euclidean distance between feature points depends only on the scale difference between the two images, and is invariant to rotation and translation, the scale factor can be estimated prior to the estimation of the other parameters. Because θ is very small, by linearly approximating $\cos\theta$ and $\sin\theta$, (8) can be rewritten as

$$\begin{pmatrix} X_{i2} \\ Y_{i2} \end{pmatrix} = \begin{pmatrix} 1 & \theta \\ -\theta & 1 \end{pmatrix}\begin{pmatrix} sX_{i1} \\ sY_{i1} \end{pmatrix} + \begin{pmatrix} \delta X \\ \delta Y \end{pmatrix}; \qquad for\ i = 1,\ldots,n \quad (10)$$

This is equivalently written as

$$A\mathbf{X} = B \quad (11)$$

where

$$A = \begin{pmatrix} sY_{11} & 1 & 0 \\ -sX_{11} & 0 & 1 \\ \vdots & \vdots & \vdots \\ sY_{n1} & 1 & 0 \\ -sx_{n1} & 0 & 1 \end{pmatrix}_{2n\times 3} \quad X = \begin{pmatrix} \theta \\ \delta X \\ \delta Y \end{pmatrix}_{3\times 1} \quad B = \begin{pmatrix} X_{12} - sX_{11} \\ Y_{12} - sY_{11} \\ \vdots \\ X_{n2} - sX_{n1} \\ Y_{n2} - sY_{n1} \end{pmatrix}_{2n\times 1}$$

This is then solved in the least square form

$$A^T A\mathbf{X} = A^T B \tag{12}$$

where A^T is the transpose of A.

After the initial four-parameter transformation is obtained using the initial matches, image O_2^k is transformed into the image N_1^k coordinate system. Area correlation matching is then applied to these images to get a new set of feature matches. This process is repeated until the transform parameter differences obtained at two successive steps are small enough.

Multi-Resolution Matching After the initial matching is achieved, the multi-resolution transform-and-correct matching process can be carried out to determine high accuracy correspondences between the two images by using the eight-parameter transform. We gave previously the plane eight-parameter transformation between two images based on their known camera models. If nothing is known about the camera models of the two images, we can use corresponding points between the two images to obtain the eight transformation parameters. From (6) and (7), we have

$$A\mathbf{X} = B \tag{13}$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{18} \\ a_{21} & a_{22} & \dots & a_{28} \\ \vdots & \vdots & \vdots & \vdots \\ a_{2n1} & a_{2n2} & \dots & a_{2n8} \end{pmatrix}_{2n\times 8} \tag{14}$$

and

$$a_{11} = X_{11}; \quad a_2 = Y_{11}; \quad a_{13} = 1; \quad a_{14} = 0;$$

$$a_{15} = 0; \quad a_{16} = 0; \quad a_{17} = -X_{11}X_{21}; \quad a_{18} = -Y_{11}X_{21}$$

$$a_{21} = 0; \quad a_{22} = 0; \quad a_{23} = 0; \quad a_{24} = X_{11};$$

$$a_{25} = Y_{11}; \quad a_{26} = 1; \quad a_{27} = -X_{11}Y_{21}; \quad a_{28} = -Y_{11}Y_{21}$$

$$\vdots$$

$$a_{2n1} = 0; \quad a_{2n2} = 0; \quad a_{2n3} = 0; \quad a_{2n4} = X_{1n}$$

$$a_{2n5} = Y_{1n}; \quad a_{2n6} = 1; \quad a_{2n7} = -X_{1n}Y_{2n}; \quad a_{2n8} = -Y_{1n}Y_{2n}$$

along with

$$\mathbf{X} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_8 \end{pmatrix}_{8\times 1} \qquad B = \begin{pmatrix} X_{21} \\ Y_{21} \\ \vdots \\ Y_{2n} \end{pmatrix}_{2n\times 1} \tag{15}$$

where n is the number of matched corresponding points.

In matrix A, if there exists a set of four points of which any three points are not collinear, the rank of the resulting matrix is 8, and (13) has a unique solution. When more than four points are used, the system is over-determined and the solution is obtained by solving

$$A^T A\mathbf{X} = A^T B \tag{16}$$

where A^T denotes the transpose of A. Special care must be taken in solving this system, which may be ill-conditioned. One solution is to use a singular value decomposition algorithm, as is done in [Tsai and Huang, 1984].

We now turn to the multi-resolution matching process. Starting from the lowest resolution images O_2^k and N_1^k and matched feature pairs, the eight-parameter transformation between O_2^k and N_1^k is first obtained by substituting matched feature pairs into (13), and solving the least square equation (16). Next, image O_2^k is transformed into the coordinate system of N_1^k. This brings the two images into closer registration. More accurate matched pairs

can therefore be obtained by area correlation on the two images. Finally, the resolution of both images is increased by one level, and the resulting images are respectively denoted by O_2^{k-1} and N_1^{k-1}. This process is repeated until the images' resolution is restored to the original level, i.e. O_2^0 and N_1^0. By then, if the difference between transformations obtained at two successive steps is small enough, the procedure is halted. Otherwise, the match-and-transform process is repeated on O_2^0 and N_1^0 until satisfactory results are obtained.

Match Verification Automatic exclusion of false matches is a key element in the success of all image registration methods. We have used three tests to exclude less reliable matches.

1. **Distance test**: The translation between the camera rotation compensated images should not be larger than a certain fraction of the image size. A valid matching pair, (X_r, Y_r) and (X_l, Y_l), should satisfy

$$\begin{cases} d_x = |X_r - X_l| \leq \lambda L_x \\ d_y = |Y_r - Y_l| \leq \lambda L_y \\ |X_r - X_l| + |Y_r - Y_l| \leq \kappa \max\{L_x, L_y\} \end{cases} \tag{17}$$

   For example, $\lambda = \frac{1}{2}$ and $\kappa = \frac{3}{2}\lambda$. L_x and L_y are the image sizes along the x and y directions, respectively.

2. **Variation test**: The translations of the correct matches should support each other, i.e.

$$|d_i - \overline{d}| \leq \mu\sigma \tag{18}$$

   where d_i is the distance between the i-th matching pair, $\overline{d}$ and σ are the mean and standard deviation of the distances for all the matched feature pairs, and μ is a threshold, for example $\mu = \sqrt{3}$ for the uniform distribution.

3. **Outlier exclusion**: The matched feature pairs should satisfy the image transformation model. Candidate matching pairs with large residual errors in (6, 7) should be excluded. This test also helps in excluding matches on building roofs.

After the 2D locations of the control points are obtained from image-to-image registration, the space resection method is applied and the camera orientation for the new image is

obtained. The new image with its internal and external camera model is then integrated into the site model.

4.2 Experimental Results

The automatic registration algorithm described above has been tested on all of the RADIUS model-board-2 images as well as on real images (Denver, CO and Ft. Hood, TX sites). The results appear to be satisfactory. Figure 3 shows the process of automatic registration of an image to the site model. Figure 3.a is an image registered to the site model, Figure 3.b a newly acquired image, Figure 3.c shows the old image finally registered to the new image, and Figure 3.d shows the result of registration of the new image to the site model. The same steps are repeated for the Denver Site image, as shown in Figure 4, as well as for the Ft. Hood site, as shown in Figure 5. Table 2 lists the results and gives comparisons between manual and fully-automatic methods.

The registration results displayed in the above images and the residual pixel errors listed in the table indicate good registration performance. Correct projection of the site model into the image subsequently allows for the application of model-based analysis techniques. The result table shows that automatic registration leads to space resection performance similar to that obtained using the manual method.

5 Detection and Counting of Vehicles in Designated Areas

The vehicle detector is used to monitor areas such as parking lots, roads or training grounds. The general vehicle detection scheme relies on contour matching using information derived from the geometric model. The procedures used to carry out vehicle detection are reported in this section.

5.1 Principle of Operation

Since we are primarily concerned with high altitude imagery in our implementation, vehicles are modeled as 3D cuboids with given width, length and height specifications. The implementation of our vehicle detector consists of a prescreener, an extractor and a verifier. The

prescreener relies on Hough transform techniques to locate areas where the centers of vehicles are likely to lie. The extractor performs template matching, and hypothesized vehicles are subsequently examined in more detail. The verifier checks for shadows to guarantee the correctness of the results. Throughout the detection procedure, 3D object models and site information (camera model, illuminant, etc.) are used. Details of the three-stage vehicle detector are now described.

Prescreener A modified generalized Hough transform (GHT) is used to locate areas corresponding to centers of candidate vehicles. The goal is to have edge pixels vote for possible loci of reference points. In our case, the reference point of an edge pixel is the center of a vehicle template which contains this pixel. Templates are the projected contours of vehicles (rectangles in a high altitude image). The displacement vector from any edge pixel to its reference point is represented using polar coordinates, indexed by the gradient direction of the edge pixel. All such displacement vectors for a given vehicle template are precomputed to form a table (see Table 3). The relevant geometry used to form the table is shown in Figure 6. The prescreening algorithm can be described as follows. Edge pixels are first extracted using the Canny edge detector. Both gradient magnitude and gradient direction are computed. A Hough table is then constructed for all vehicle templates. An accumulator array of possible reference points is created, and initialized to zero. For an edge pixel (x, y), we use its gradient direction ϕ to retrieve its associated displacement vectors from the Hough table and derive the positions of all possible reference points. Therefore, each edge pixel will cast its vote for all reference points (x_c, y_c), where

$$\begin{aligned} x_c &= x + r(\phi)\cos[\alpha(\phi)] \\ y_c &= y + r(\phi)\sin[\alpha(\phi)] \end{aligned}$$

The result of this step is a set of hypothesized vehicle centers, $\mathcal{L}$. Each point in the set $\mathcal{L}$ represents the center of a possible vehicle which has a match ratio (between its boundary contour and edge map) above a determined conservative threshold. Since we adopt a thresholding scheme instead of searching for peaks in Hough space, we proceed with the detailed examination of vehicle contours.

Extractor We subsequently apply local template matching using "rubber-band" templates. This stage consists of four steps: (1) determining the positions and orientations of 3D vehicle models; (2) computing contours of 3D models for template matching; (3) computing textural features and combining the features into a discrimination statistic that measures how "target-like" the detected object is; and (4) clustering into geometrically consistent objects. The algorithm first derives the positions and orientations of 3D models. Centers of 3D models need to be consistent with the back-projections of points in $\mathcal{L}$. Their orientations are also constrained to lie along the dominant direction of the active area. Vehicle templates are computed from their 3D model.

For template matching purposes, rubber-band templates are used instead of tolerance band templates. The major disadvantage of template matching with tolerance bands is that it repetitively counts all pixels within tolerance bands. On the other hand, a rubber-band template is deformable; therefore, it will not overrate pixels clustered along the directions normal to the template contours. In addition to this, the degree of matching is evaluated separately on the vehicles' boundaries, so that to qualify as a vehicle, the target needs to have a sufficient support boundary on both sides of the vehicle.

All points in $\mathcal{L}$ at which templates sufficiently match sets of edge pixels are declared centers of candidate targets. These points need to be further examined so that vehicles centered at these locations are geometrically consistent. The selection of competing points is carried out using textural information. During this stage, three textural features are computed for each candidate: (1) the mean value, (2) the standard deviation, and (3) the maximum of the intensity distribution of the pixels lying within the target-sized template. When two hypothesized vehicles are geometrically inconsistent, e.g. overlapping, the one having higher quality measure survives. The quality measure is a weighted sum of the quality of contour matching and the similarity of the statistics to the reference statistics. The reference statistics are computed based on those of highly matched targets. The similarity of statistics is calculated by inverting a quadratic distance measure. At this step, most of the false alarms are rejected due to either a large quadratic distance or bad contour quality.

Verifier The verifier stage primarily involves shadow verification. Recognition from contours usually suffers from the ambiguity existing between similar 2D shapes arising from the contours of different 3D objects. One way to avoid this problem is to use shadow information. Since shadows only accompany 3D objects, false alarms can be minimized by checking for the existence of a shadow within a predicted region. Shadows have high contrast and darker intensity than the background, and their size can be predicted using the vehicle orientation, viewing direction and illumination direction. In our implementation, we use an enhancement process and a region growing process which segments an image into regions based on the intensity differences between connected components [Rodriguez, 1994]. The resulting region segments are used to verify the existence of vehicle shadows. Detection is declared if there exist segments that comprise a region satisfying the following constraints. **Size** constraint: The size is computed by projecting the object model to the ground plane. The visible portion of the shadow from the viewing direction is then computed. **Intensity** constraint: The intensity of the shadow region is both homogeneous and dark. In other words, both the deviation and the mean value of the intensity distribution are small. **Position** constraint: The shadow position is consistent with the illumination direction. **Shape** constraint: The shadow can be either on one or two consecutive sides of the vehicle.

5.2 Experiments

The above vehicle detection scheme has been integrated into the RCDE exploitation platform. Results of its application to both parking lots and roads are shown on real operational imagery. Figure 7 shows a typical example of vehicle detection in a parking lot in the Martin Marietta, Denver, CO site. (a) shows an image to be exploited; (b) shows the area corresponding to the parking lot of interest, delineated from the region information in the site model; and (c) shows the detected vehicles. For vehicle detection in the parking area, we used information about the orientation of the area to constrain the possible vehicle parking directions. Figure 8 and Figure 9 show detection examples on four other real or synthetic test sites: the MB2, Denver, CO, Ft. Hood, TX, and Schenectady, NY, sites.

6 Monitoring Construction Activities

In this section, we discuss the subsystem for monitoring construction activities.

6.1 Principle of Operation

The process of monitoring construction activities relies on an explicit geometric description of the object of interest and on recognition by parts. It therefore relies heavily on site model information. An example of this technique applied to the detection of cylindrical structures is presented below. Details about the features used for target representation, the feature extraction process, and the hypotheses generation/verification schemes are also given.

We use a hierarchically parameterized object model to incorporate knowledge from the site model and the image acquisition conditions into the low level processes. A 3D cylinder can be geometrically described by its height, the radius of its cross-section and the center of its base, denoted by h_{3d}, r_{3d} and $(x_w, y_w, 0)$. Using the camera model resected from image-to-site-model registration, we can convert the 3D geometric model to a 2D model including an ellipse, a pair of parallel lines, and some geometric relations among them. A 3D cylinder is declared to be detected if we can extract all the necessary features, and if among these features, all the derived geometric constraints are satisfied. In the following paragraphs, more details are given on the feature extractors implemented to detect cylindrical objects.

Ellipse Detection In order to extract ellipses, we first apply a modified Canny edge detector [Venkateswar and Chellappa, 1992] to compute the edge map and gradient directions of edge pixels. Instead of directly using Hough transforms [Ballard, 1981; Shapiro, 1978], which require a prohibitive amount of memory, we have implemented a two-stage template matching scheme for ellipse detection. In the first stage, edge templates are used to determine possible candidate centers. In the second stage, gradient direction templates are used to re-inspect the selected candidate center points. A gradient direction template can be formulated in terms of the camera rotation angles and the the direction from the center to the corresponding pixel. In our implementation of template matching, a three pixel wide tolerance band on the edge template is allowed to accommodate slightly misplaced pixels.

For an edge pixel to be a supporting pixel, the pixel must fall within the tolerance band and also have a gradient direction consistent with the gradient template. We accept those candidates whose consistency scores are above the high threshold for an ellipse. For candidates whose consistency scores fall between the high and low thresholds, we further apply a radius histogram test; if we plot a histogram of intensity as a function of distance from the candidate center, a steep slope should appear at the radius of the 3D circle.

Detection of Cylinder Body The silhouette of a vertical cylinder body is always projected as a pair of parallel lines whose directions are along the camera viewing direction. We first link edges into line segments using a line linking process which scans the edge map, and label the edge pixels according to some predefined templates. After the labeling process, edge pixels are grouped into line fragments and, furthermore, small collinear line fragments are merged into long straight lines.

To detect evidence corresponding to the body of a cylindrical object, we first need to locate the parallel lines which are oriented consistently with the projected silhouette of a cylinder body according to the camera parameters. Consider two lines, shown in Figure 11 as L_i and L_j, forming a pair of parallel lines, $\text{Para}_{i,j}$, and satisfying the orientation constraint. For them to become evidence of the body of a cylinder, two more geometric properties need to be tested:

1. distance: $(d_{ij} + d_{ji})/2 \in (r_{\min}, r_{\max})$
2. overlap: $(l_i + l_j)/(2 \times l_{i,j}) < 0.5$

where d_{ij} is the distance from M_i, the midpoint of L_i, to line L_j; l_i and $l_{i,j}$ are the lengths of the line segments L_i and VL_{ij}, respectively; $r_{\min}$ and $r_{\max}$ are the width constraints on the cylinder's body, derived from the radius of the cylinder's cross section and the camera parameters.

Hypothesis Generation As shown in Figure 12, we will hypothesize the existence of a cylinder if we can group a detected ellipse C_k with supporting parallel line pairs, $Para_{i,j}$, and if they satisfy the following constraints: First, the distance from O_k to P_s should be less

than the projected height of the cylindrical object. Second, the direction of vector $\overline{M_{ij}O_k}$ should be the same as the direction of the projection of an upward vertical vector in the 3D world. After the grouping, the quality of the grouping is evaluated from the quantity

$$H(C_k, \text{Para}_{i,j}) = \sum_{l=1}^{3} w_l \times H_l(C_k, \text{Para}_{i,j}) \tag{19}$$

where

$$w_1 = w_2 = w_3 = \frac{1}{3} \tag{20}$$

$$H_1(C_k, \text{Para}_{i,j}) = \frac{d1}{R_k} \tag{21}$$

$$H_2(C_k, \text{Para}_{i,j}) = \frac{|R_k - d2/2|}{\epsilon_{\text{width}}} \tag{22}$$

$$H_3(C_k, \text{Para}_{i,j}) = \frac{|\overline{O_k P_e}|}{|\overline{O_k P_s}|} \tag{23}$$

and $d1$ denotes the distance from O_k to $VL_{i,j}$ and $d2$ the distance between L_i and L_j. If $H(C_k, \text{Para}_{i,j})$ is less than a threshold, a cylindrical object located at the corresponding position is hypothesized.

6.2 Experiments

In our implementation, a hypothesis needs to satisfy the following three tests before being accepted as a detected cylindrical object. These tests are used to check for more support from the original edge map, shadow information, and intensity distribution. First, we fit a model and check its consistency with the original edge map. Second, using the illumination information, we delineate a region where the shadow of the proposed cylinder might appear. Within that region, we look for a supporting shadow (a homogeneously dark region) bounded by a pair of parallel lines. Lastly, the intensity variations within the ellipse and the region bounded by the parallel lines should be much smaller than the intensity variation in the image.

Figure 13 shows an example of cylindrical object detection: (a) displays the detected edges overlaid on the original image, (b) the result of line linking, (c) the detected ellipses, and (d) the final result. Figure 14 shows another example that has one newly added cylindrical object. (a) displays the detected edges overlaid on the original image and (b) shows the

final result. Change can be inferred from direct comparison with the objects detected in the previous image.

7 System Integration

The detection modules described above as well as the automatic image registration module have been developed as part of the RADIUS project. The principal goal of the RADIUS project is the development of IU capabilities focused on the needs of IAs. The RADIUS Common Development Environment (RCDE) platform jointly developed by SRI and Lockheed Martin implements basic photogrammetric procedures and low-level image processing and manipulation functions; it also allows for the creation and manipulation of CAD objects used to model site objects. These objects are stored in a database. This platform provides a common development environment on which RADIUS related algorithms can be developed and tested. It also provides a system from which an operational RADIUS testbed system is derived. Finally, this platform offers many functionalities that are exploited by the context-based modules we have developed, the most important being the ability to create, store and retrieve object, image or site information that is necessary in context-based exploitation. Most of the modules we have presented here have been ported to, or developed around, the RCDE system.

A typical exploitation scenario making use of the RCDE basic functionalities is conducted as follows. A site model containing permanent objects is constructed. The prerequisites of the site model construction are the availability of several partially overlapping images of the same site. Multiple views of the same site objects are indeed necessary to disambiguate the object positions and dimensions. When images are simultaneously displayed, a shared 3D local world coordinate system is defined for all images, and registration is carried out. As previously mentioned, in the usual photogrammetric applications, images are provided with a set of interior as well as exterior orientation parameters with associated covariance matrices. These are usually supplied in a TEC header format. These initial orientation parameters are refined by using our registration methods along with RCDE-supplied resection tools, after control point objects have been defined (these are points whose 3D positions

are known). Control points are used along with their identified image positions to refine camera orientation. Conjugate points are determined, either manually, or using one of our automated procedures. Single or multiple image resection is then accomplished.

When image orientation has been registered to a common local vertical coordinate system, site construction can be initiated. 3D generic objects are selected, created, and modified. Objects of primary importance in our case are roads, buildings, delineated training areas, and parking lots. Alternate objects are rivers, ships, etc. Objects are manipulated and modified interactively by the operator, or defined through dialog windows. This site-generation procedure is carried out until a satisfactory site model has been constructed from all viewpoints.

When a new image is acquired, it is registered to the existing site model, and is thereupon exploited. Site models are not static. While the detection algorithms reported here exploit site model information to detect changes, they can be used, in turn, to verify and detect discrepancies, and to refine and update site model information. These algorithms are not indiscriminately applied to all incoming images and all areas. Instead, the IA specifies which areas should undergo special scrutiny.

One operational procedure of particular importance, in driving the exploitation of images, is determined by the concept of quick-look (QL) [Bailey et al., 1994]. In the QL mode, only small areas (usually corresponding to functional regions) are processed in batch over multiple images; historical comparisons are then carried out to determine trends and evolutions. Only a limited number of areas in the site are exploited: usually small changes in these areas are significant. Changes must, however, be detected in a timely fashion over a large set of images. Active QL areas are then prioritized for further and finer exploitation.

In the QL scenario, areas supporting the presence of convoys are brought to the attention of the IA. The IA may then trigger a specialized vehicle detector module, according to the type of ROI (road or parking). Exact vehicle counts are then reported to the analyst.

The vehicle detector has been fully integrated into the RCDE platform. It has been implemented in C, and later reimplemented in C++. The RCDE platform is instead implemented in Lucid Lisp. Part of the vehicle detection process, consisting mostly of the interface, is implemented in Lisp. The Lisp/C interface of the RCDE enables inter-process

communication between the C/C++ modules and the Lisp process. The integrated vehicle detector has been tested by several institutions including GE, Lockheed-Martin and the National Exploitation Laboratory. Its interface is currently being tailored to the specific needs of IAs.

Figure 10 shows an example of a dialog window for the integrated implementation of the vehicle detector module. After vehicles have been detected, the result is stored in the database for later inspection.

8 Conclusion

We have described the recent work performed under the RADIUS project for detection and counting, monitoring, and automatic positioning. The principal activities reported are: (a) vehicle related activities and (b) construction related activities. Our approaches make extensive use of site and object model information and rely on the application of context based IU algorithms. Special emphasis has been placed on the integration with, or development of, these models around the RCDE platform. The performance of the described algorithms has been tested on operational imagery.

References

[ARPA, 1993] ARPA. Radius testbed system operations concept, July 1993.

[Bailey et al., 1994] J.S. Bailey, M.D. Kelly, and J.D. Sargent. Quick-look: A new way to prioritize imagery for exploitation. In *Proc. ARPA Image Understanding Workshop*, pages 247–249, 1994.

[Ballard, 1981] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122, 1981.

[Beveridge and Riseman, 1992] J.R. Beveridge and E.M. Riseman. Hybrid weak-perspective and full-perspective matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 432–438, 1992.

[Collins et al., 1993] R.T. Collins, A.R. Hanson, E.M. Riseman, and Y.Q. Chen. Model matching and extension for automated 3D site modeling. In *Proc. ARPA Image Understanding Workshop*, pages 197–204, April 1993.

[Faugeras, 1993] O. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint.* M.I.T. Press, Cambridge, MA, 1993.

[Huertas and Nevatia, 1988] A. Huertas and R. Nevatia. Detecting buildings in aerial images. *Computer Vision, Graphics and Image Processing*, 41:131–152, 1988.

[Manjunath, 1992] B.S. Manjunath. A feature-based approach to face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1992.

[McKeown, 1990] D.M. McKeown. Toward automatic cartographic feature extraction. In *Mapping and Spatial Modeling for Navigation*, pages 149–180. Springer-Verlag, Berlin, Germany, 1990.

[Methley, 1986] B. Methley. *Computational Models in Surveying and Photogrammetry.* Blackie, London, England, 1986.

[Moffitt and Mikhail, 1980] F. Moffitt and E.M. Mikhail. *Photogrammetry.* Harper and Row, New York, NY, 1980.

[Rodriguez, 1994] C. Rodriguez. An appearance-based approach to object recognition in aerial images. Master's thesis, Department of Computer Science, University of Maryland, College Park, MD, 1994.

[Shapiro, 1978] S.D. Shapiro. Feature space transforms for curve detection. *Pattern Recognition*, 10:129–143, 1978.

[SRI, 1993] SRI International and Martin Marietta. *RADIUS Common Development Environment: User's Manual*, July 1993.

[Strat, 1995] T.M. Strat. Integrating IU Algorithms in the RADIUS HUB. Software documentation, 1995.

[Strat and Climenson, 1994] T. Strat and W.D. Climenson. Site model content. In *Proc. ARPA Image Understanding Workshop*, pages 277–285, 1994.

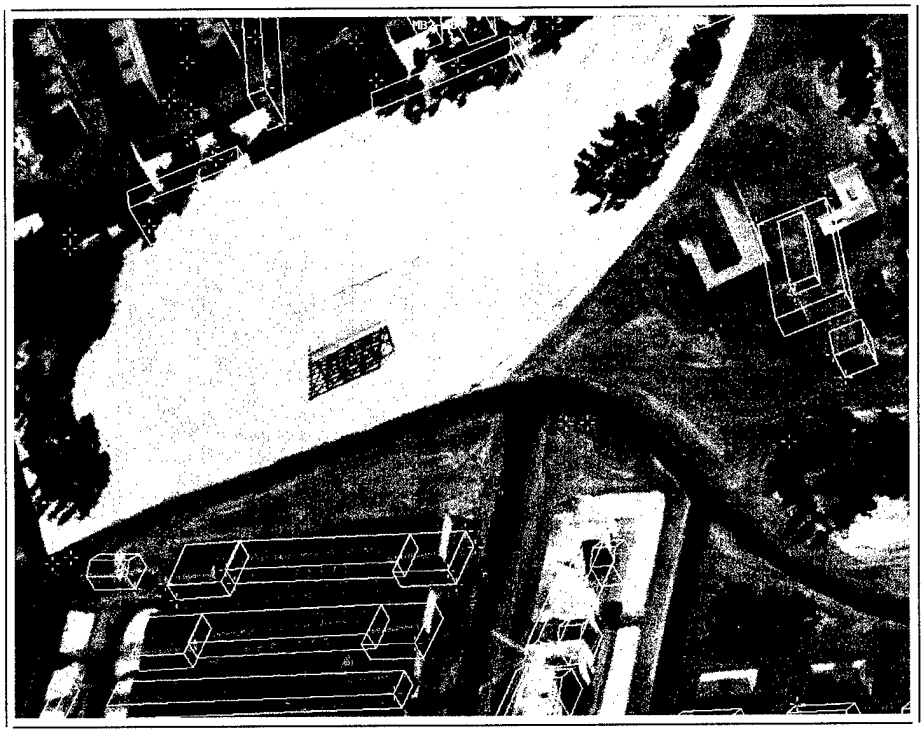
[Tsai and Huang, 1981] R.Y. Tsai and T.S. Huang. Estimating three dimensional motion parameters of a rigid planar patch. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 29:1147–1152, 1981.

[Tsai and Huang, 1984] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:13–27, 1984.

[Venkateswar and Chellappa, 1992] V. Venkateswar and R. Chellappa. Extraction of straight lines in aerial images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:1111–1116, 1992.

[Wolberg, 1988] G. Wolberg. *Digital Image Warping.* IEEE Computer Society, Los Alamitos, CA, 1988.

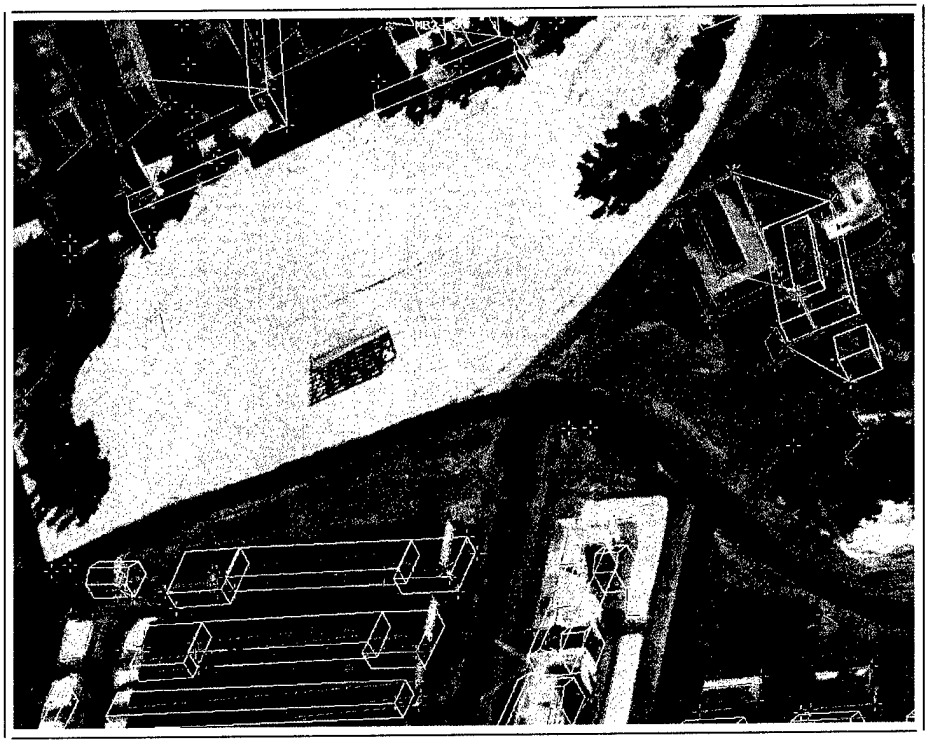
[Zheng and Chellappa, 1993] Q. Zheng and R. Chellappa. A computational vision approach to image registration. *IEEE Trans. on Image Processing*, 2:311–326, 1993.



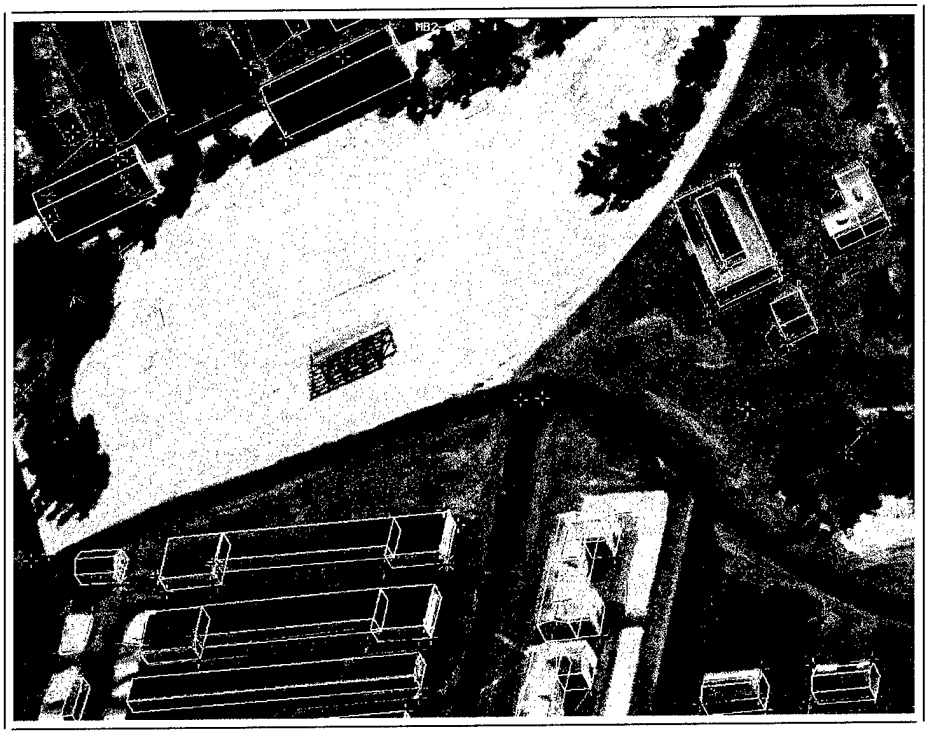
(a) Initial projection of control points on the new image

(b) Corners extracted from the new image

Figure 1: Semi-automatic site model registration of a new image



(a) Operator-guided matching of control points with the detected corners

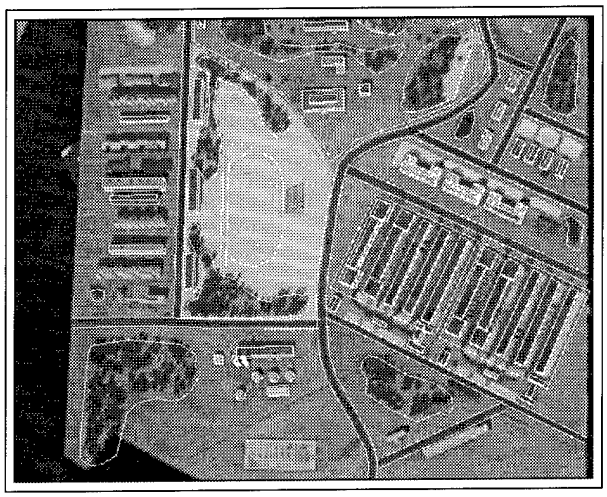


(b) Registration result

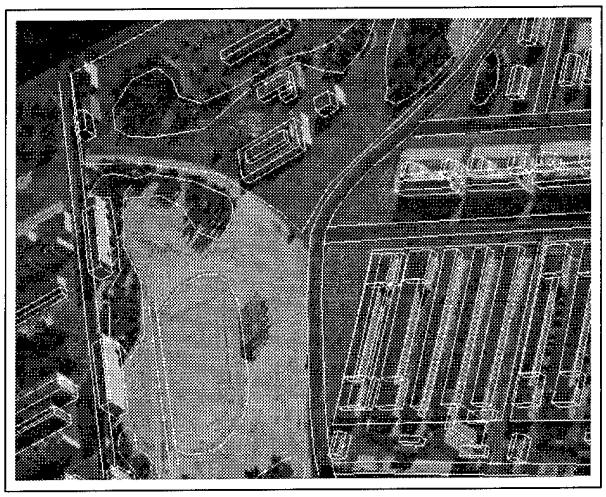
Figure 2: Semi-automatic site model registration of a new image

Table 1: Comparison of manual and semi-automatic registration results

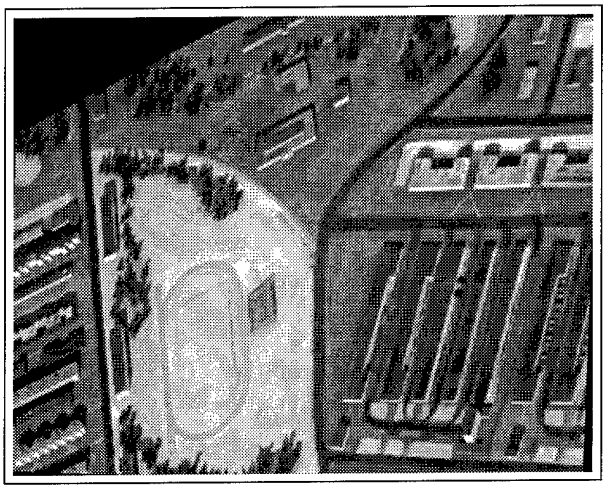
	manual			semi-automatic		
image name	exterior orientation $x_0, y_0, z_0, \omega, \phi, \kappa$	control points	RMS residuals (pixels)	exterior orientation $x_0, y_0, z_0, \omega, \phi, \kappa$	control points	RMS residuals (pixels)
mb2-m2	6716.1632 5325.5764 7431.4332 –0.5639 0.5941 1.4364	5	2.94	6715.9733 5350.7008 7379.8148 –0.5695 0.5956 1.4399	9	2.49
mb2-m16	5682.9202 7258.8355 7338.3312 –0.7130 0.4388 3.1294	5	2.14	5772.3348 7219.4180 7297.8965 –0.7131 0.4482 3.1285	10	1.94
mb2-m32	7083.5399 –4283.0598 7557.5014 0.5646 0.5822 –0.1699	6	2.79	7079.3356 –4306.3963 7592.9550 0.5648 0.5800 –0.1708	8	1.60
mb2-m39	3154.7725 –4834.7719 8997.9824 0.5381 0.1992 3.0173	8	2.56	3062.8568 –4889.9023 8935.5940 0.5458 0.1914 3.0204	7	1.81



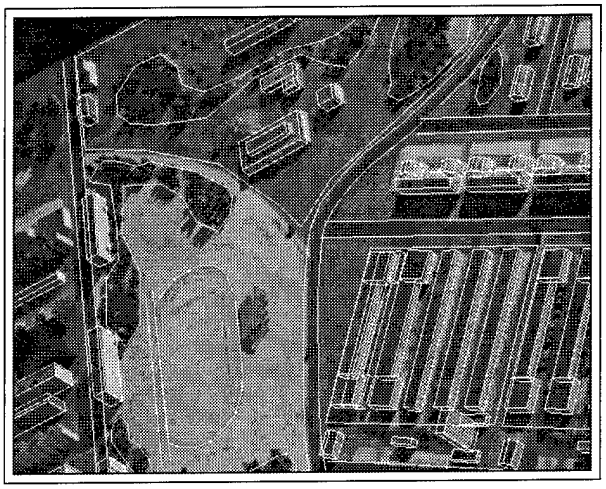
(a) Old registered image (m34)



(b) New image with approximate orientation (m2)

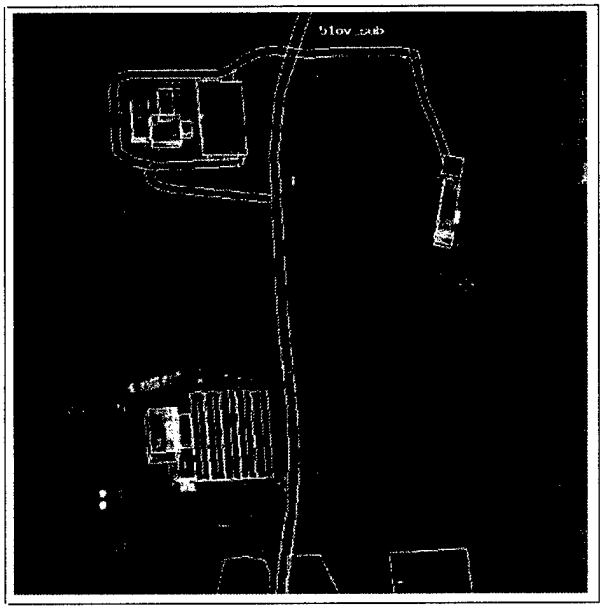


(c) Old image registered to new image



(d) Site model registered on new image

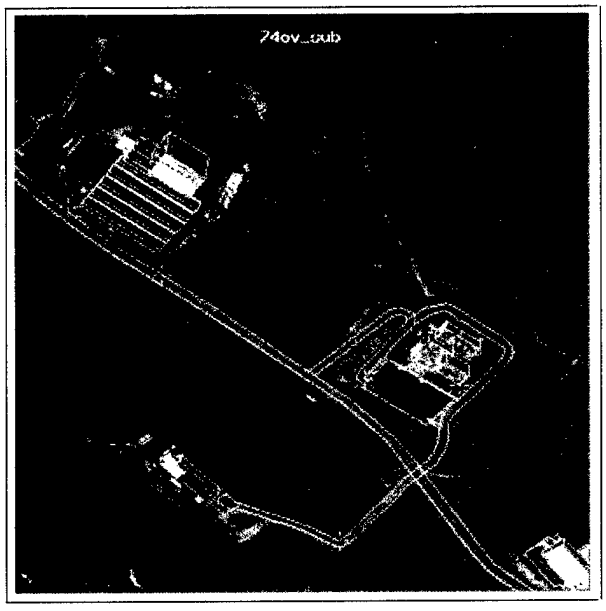
Figure 3: Automatic image-to-site-model registration



(a) Old previously registered image

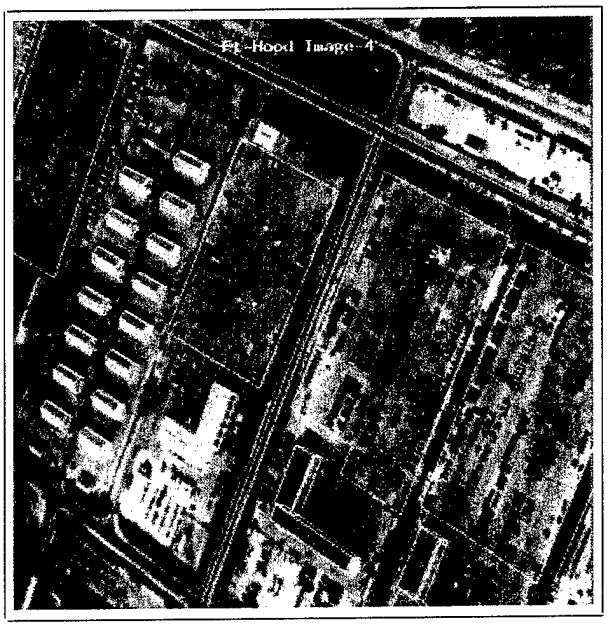
(b) New image to be registered

(c) Old image registered to new image domain

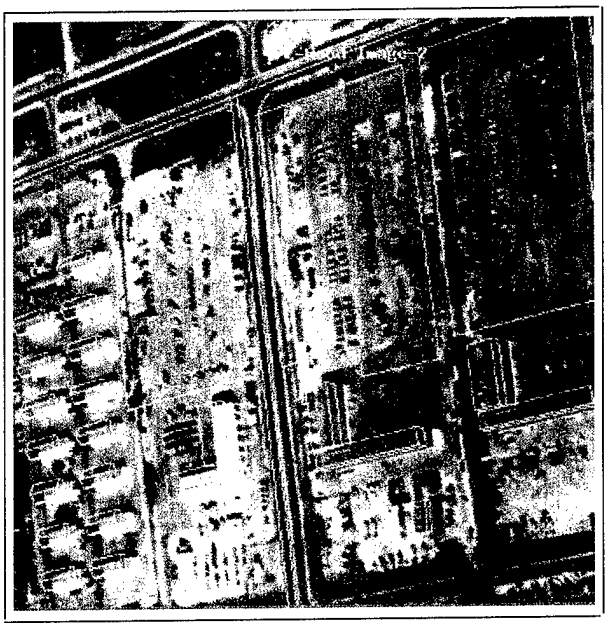


(d) Site model registered into new image

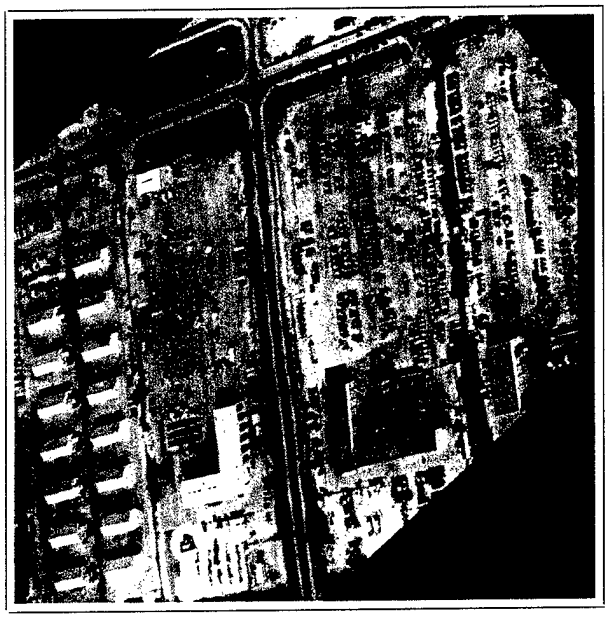
Figure 4: Automatic image-to-site-model registration



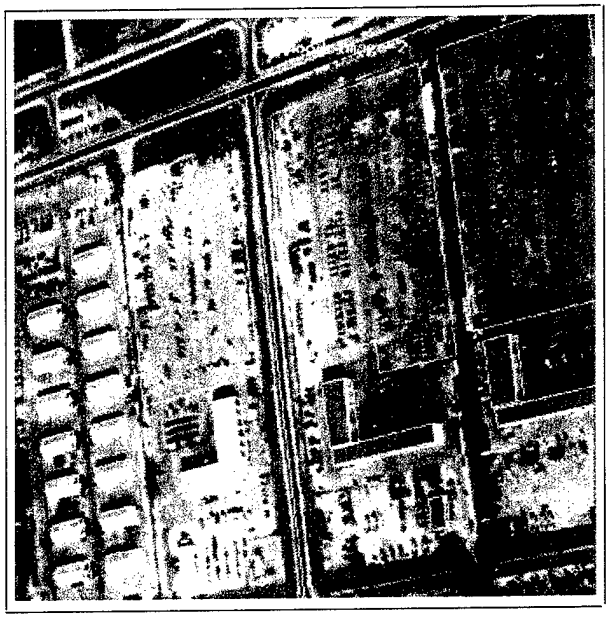
(a) Old previously registered image



(b) New image to be registered



(c) Old image registered to new image domain



(d) Site model registered into new image

Figure 5: Automatic image-to-site-model registration

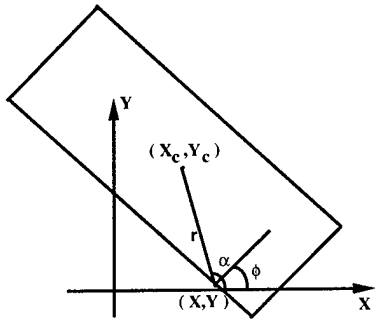


Figure 6: Geometry for reference point

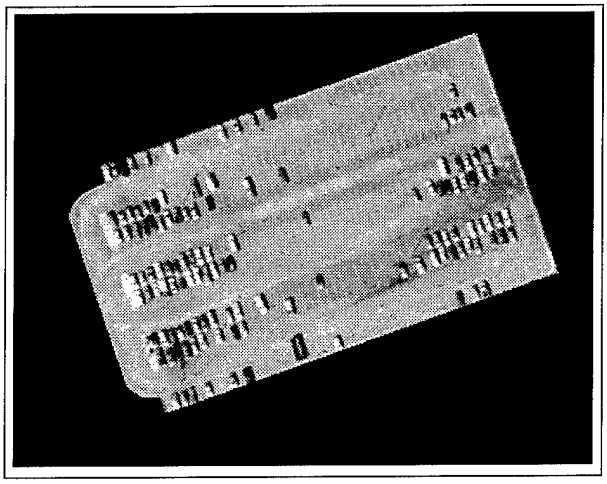
Table 2: Comparison of manual and automatic registration results

	manual			automatic		
image name	exterior orientation $x_0, y_0, z_0, \omega, \phi, \kappa$	control points	RMS residuals (pixels)	exterior orientation $x_0, y_0, z_0, \omega, \phi, \kappa$	control points	RMS residuals (pixels)
mb2-m2	6716.1632 5325.5764 7431.4332 −0.5639 0.5941 1.4364	5	2.94	6723.6983 5331.4326 7505.3336 −0.5597 0.5911 1.4309	27	1.72
mb2-m16	5682.9202 7258.8355 7338.3312 −0.7130 0.4388 3.1294	5	2.14	5681.8459 7270.9673 7405.4549 −0.7089 0.4357 3.1262	61	0.79
mb2-m32	7083.5399 −4283.0598 7557.5014 0.5646 0.5822 −0.1699	6	2.79	7084.4138 −4323.3061 7645.7963 0.5631 0.5773 6.1131	50	0.83
74ov_sub	304.2065 −1890.8012 4384.2044 0.6980 −0.0424 3.0555	4	0.16	221.0623 −1916.2258 4361.7590 0.7076 −0.0674 3.0652	33	2.58
ft-image-2	-2106.1978 5837.0582 3974.0980 0.1685 0.4727 −0.3453	8	1.26	-2098.20 5830.06 3980.10 0.1685 0.4727 −0.3452	38	2.06

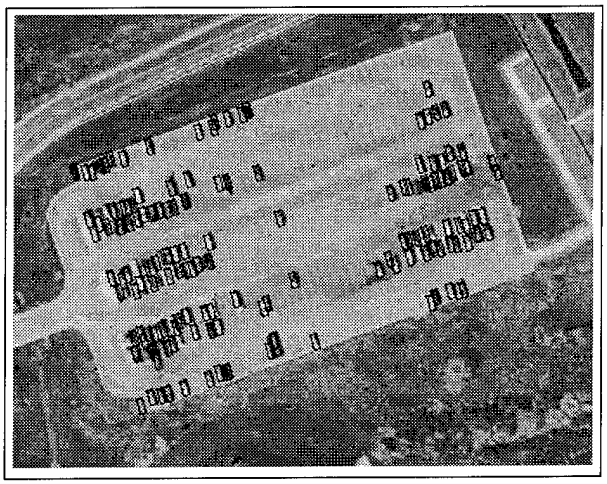
Table 3: Indexed table for reference points

Gradient direction of edge point	Set of radii r^k where $r = (r, \alpha)$
ϕ_1	$r_1^1, r_2^1, \ldots, r_{m_1}^1$
ϕ_2	$r_1^2, r_2^2, \ldots, r_{m_2}^2$
ϕ_3	$r_1^3, r_2^3, \ldots, r_{m_3}^3$
$\vdots$	$\vdots$
ϕ_{n-1}	$r_1^{n-1}, r_2^{n-1}, \ldots, r_{m_{n-1}}^{n-1}$
ϕ_n	$r_1^n, r_2^n, \ldots, r_{m_n}^n$

(a) An image to be exploited

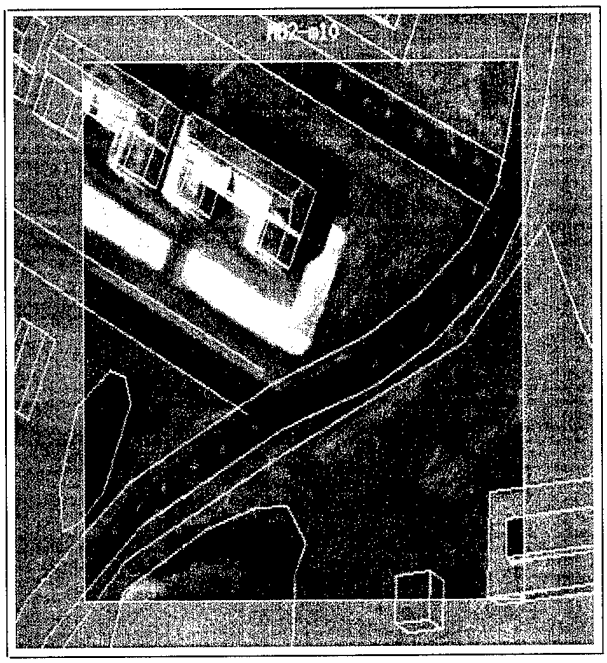


(b) Region of interest

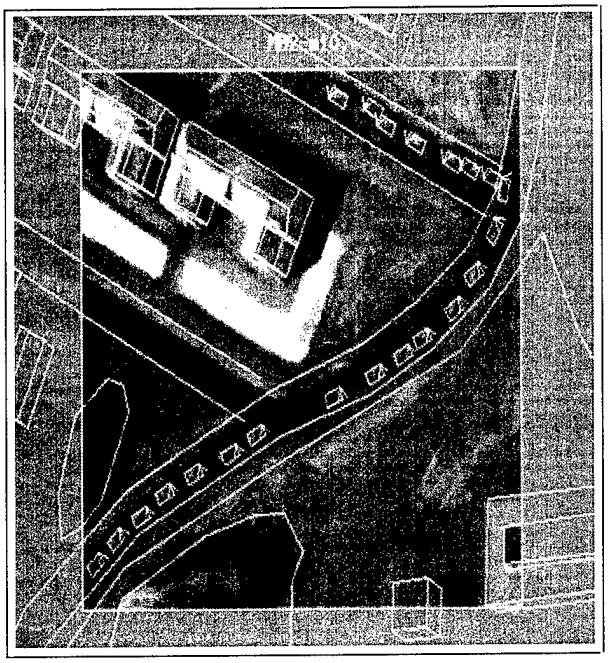


(c) Vehicles detected in the parking area

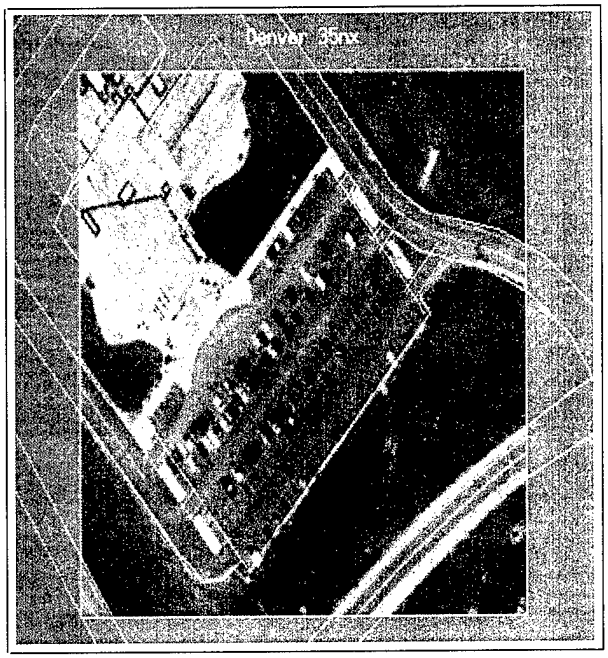
Figure 7: Vehicle detection in a parking area



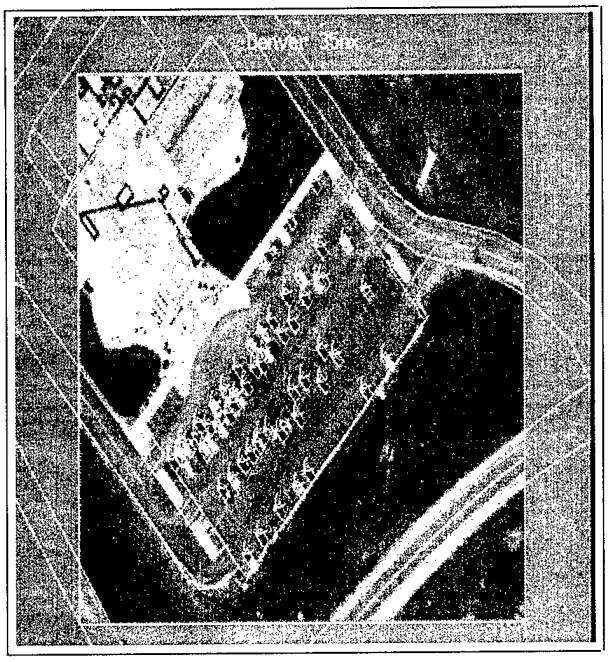
(a) ROI of model board image M10



(b) Detected vehicles superimposed on original image

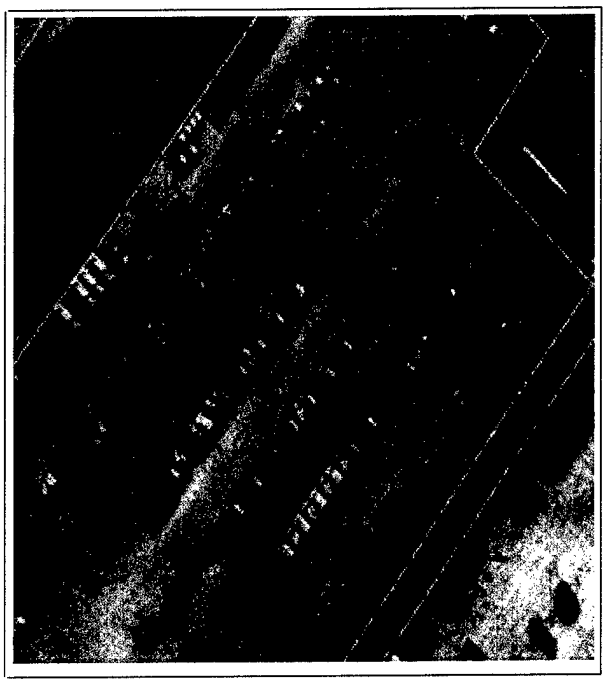


(c) ROI of Denver image

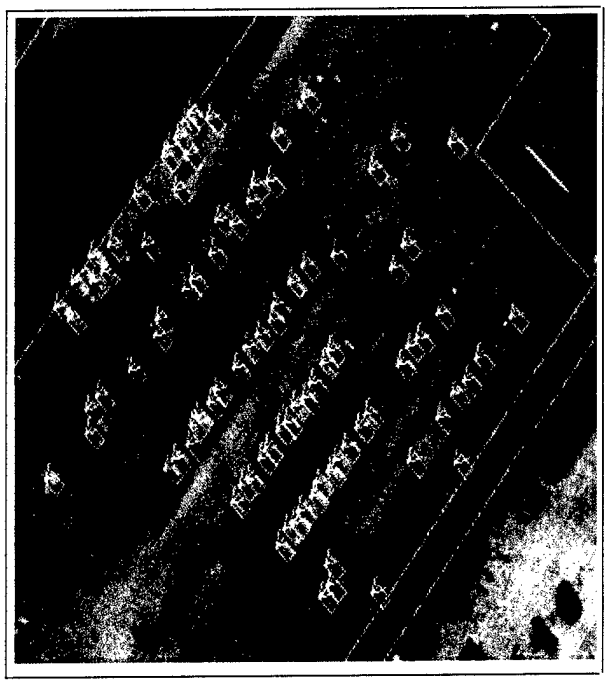


(d) Detected vehicles superimposed on original image

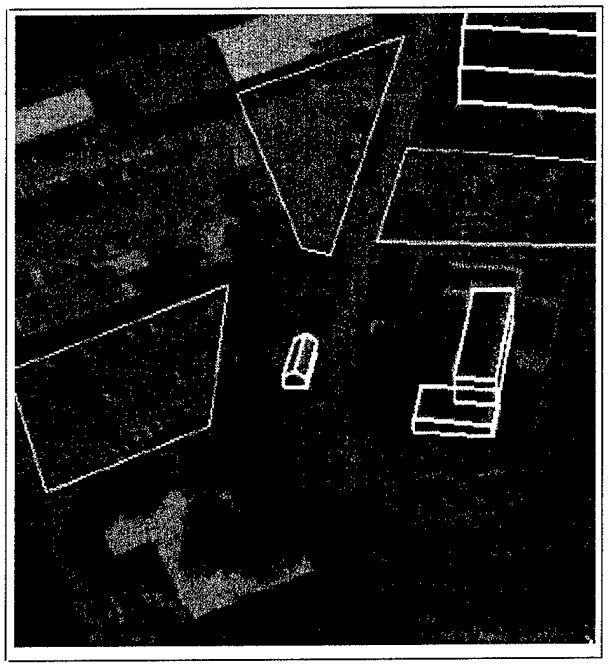
Figure 8: Model supported vehicle detection



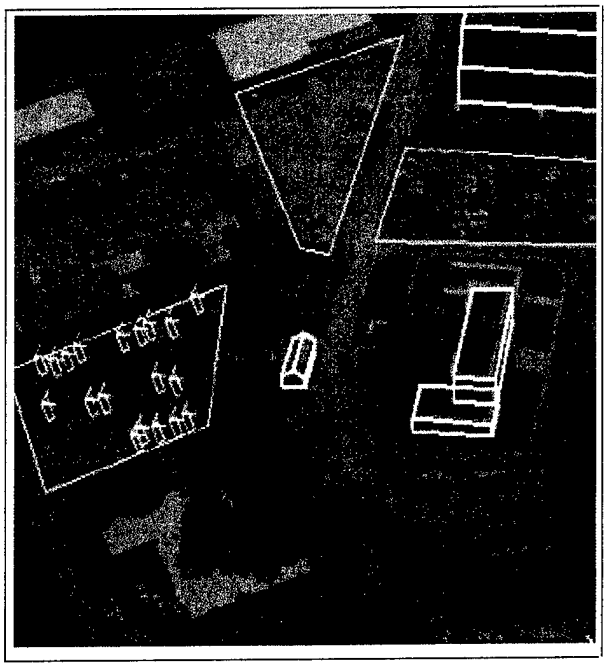
(a) ROI of Ft. Hood image



(b) Detected vehicles superimposed on original image



(c) ROI of Schenectady image



(d) Detected vehicles superimposed on original image

Figure 9: Model supported vehicle detection

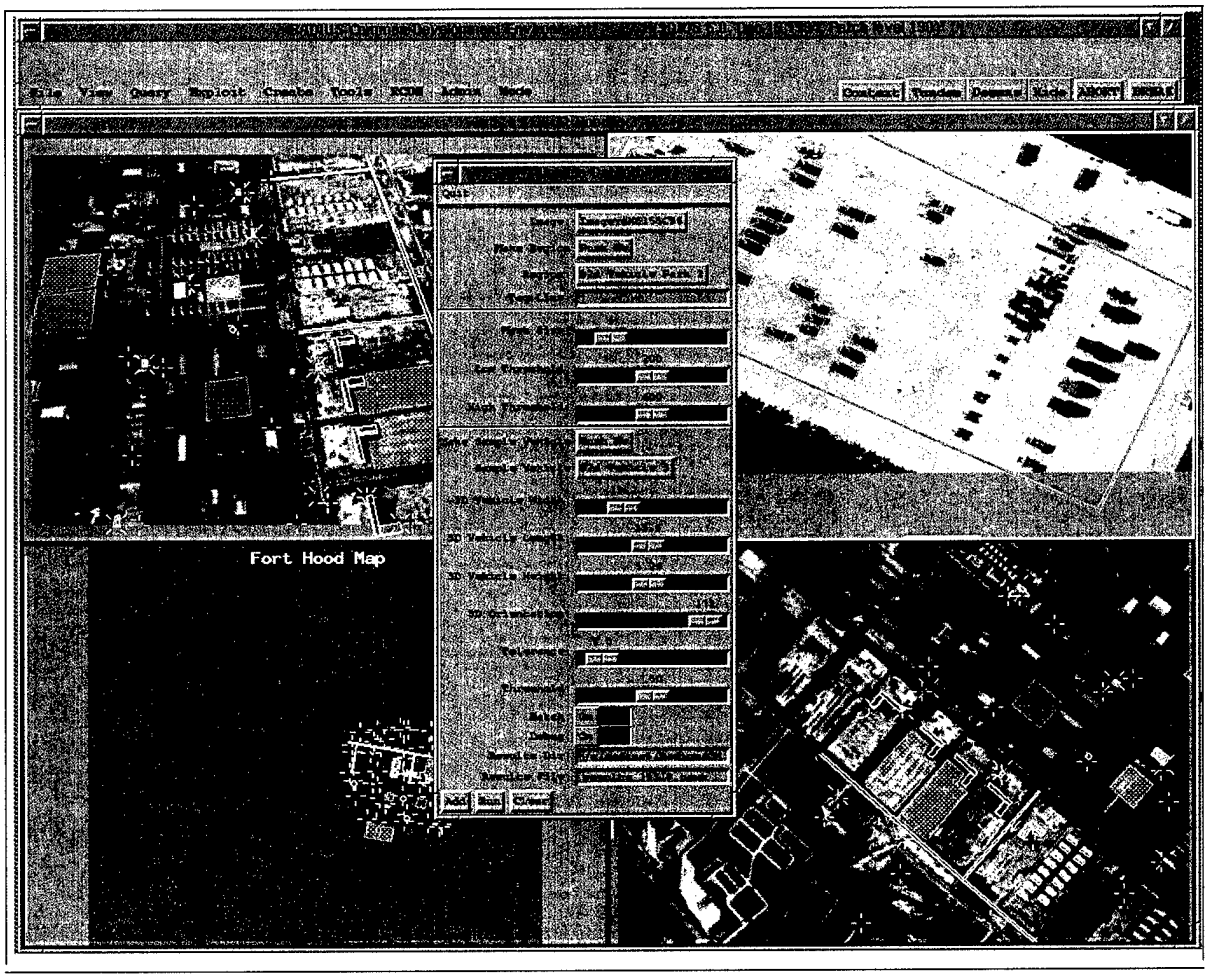


Figure 10: Integrated vehicle detector

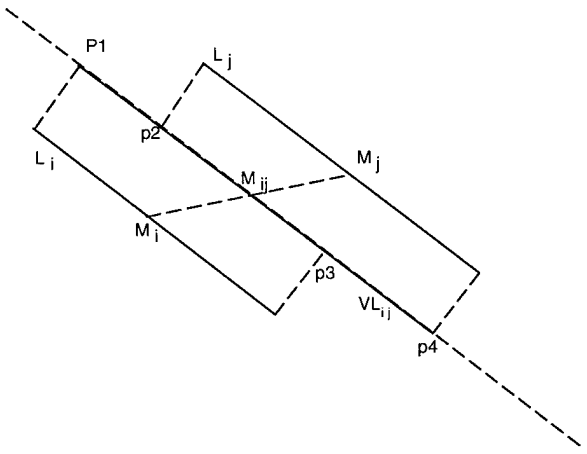


Figure 11: Line grouping

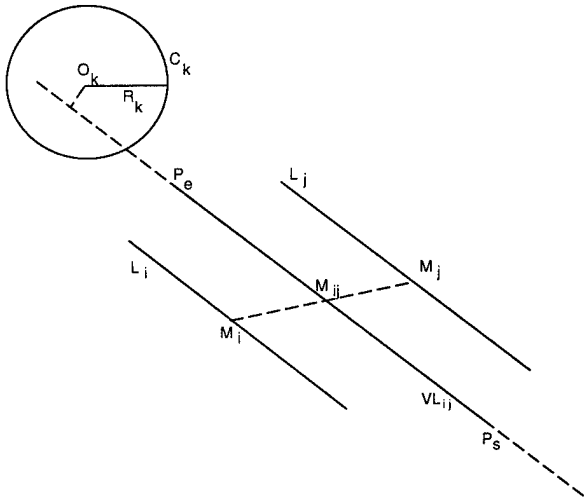
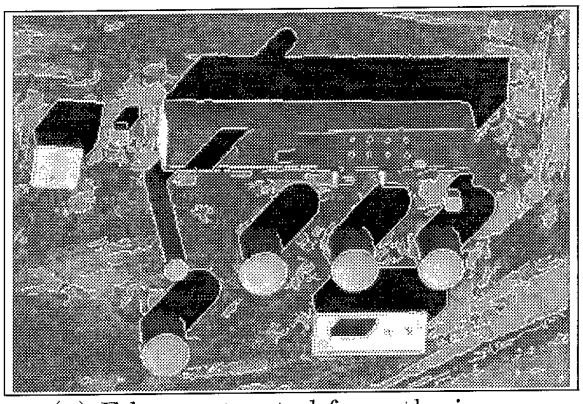
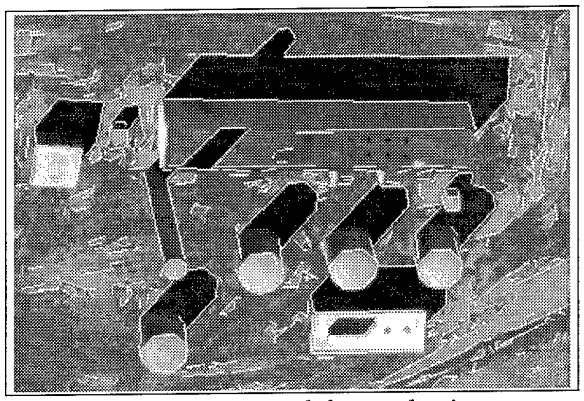


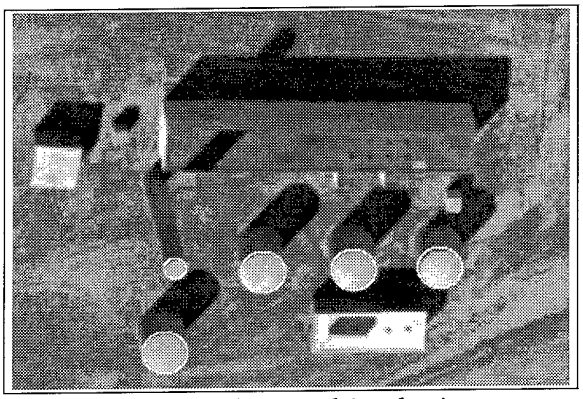
Figure 12: Feature grouping



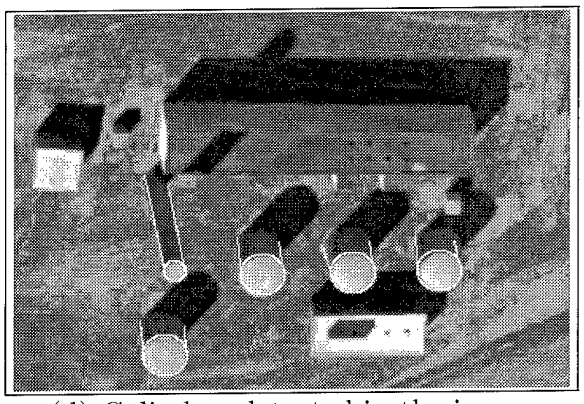
(a) Edges extracted from the image



(b) Lines extracted from the image

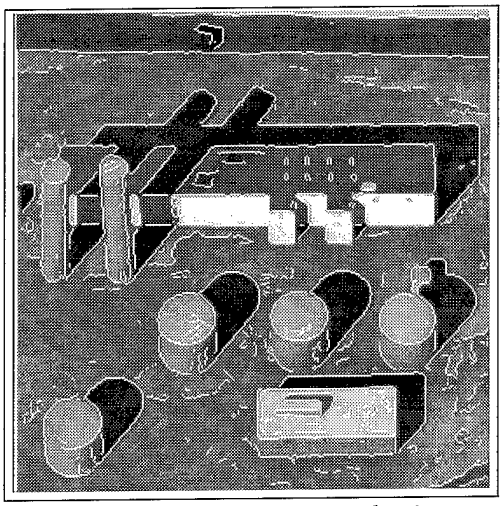


(c) Ellipses detected in the image

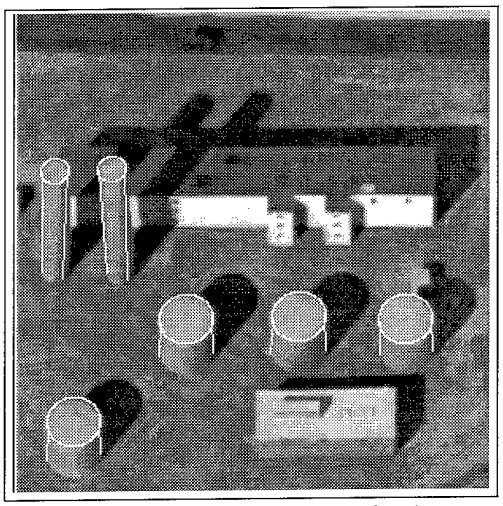


(d) Cylinders detected in the image

Figure 13: Construction detection



(a) Edges extracted from the image



(b) Cylinders detected in the image

Figure 14: New construction detection